AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A method for testing a processor including an execution stage comprising:

generating a neutral instruction that causes, when executed, an architectural state value for said processor to be ascertained;

providing said neutral instruction to said execution stage of said processor; and

executing said neutral instruction to ascertain ~~an~~ said architectural state value ~~for said processor~~.

2. (Original) The method of claim 1 wherein said neutral instruction is generated when a plurality of instructions are generated by a compiler.

3. (Original) The method of claim 1 wherein said neutral instruction is generated by a No-operation (NOP) pseudo-random generator.

4. (Original) The method of claim 3 wherein the execution of said neutral instruction causes said processor to access a value stored in a register in said processor.

5. (Original) The method of claim 1 wherein the execution of said neutral instruction causes said processor to access a value stored in a register in said processor.

6. (Original) The method of claim 1 wherein said neutral instruction is generated by a post-processor device.

7. (Currently Amended) A system for testing a processor including an execution stage comprising:

comparison logic coupled to the execution stage of said processor, wherein said execution stage is to execute a neutral instruction that is to cause, when executed, an architectural state value for said processor to be ascertained ~~to ascertain an architectural state value for said processor~~.

8. (Original) The system of claim 7 wherein said neutral instruction is generated by a compiler.

9. (Original) The system of claim 7 further comprising:

a No-operation (NOP) pseudo-random generator coupled to the execution unit of said processor to generate said neutral instruction.

10. (Original) The system of claim 9 wherein the processor includes a register and the execution of said neutral instruction causes said processor to access a value stored in the register in said processor.

11. (Previously Presented) The system of claim 10 wherein said neutral instruction includes ORing the contents of said register with itself.

12. (Original) The system of claim 10 wherein said neutral instruction includes ANDing the contents of said register with all binary 1 values.

13. (Original) The system of claim 10 wherein said neutral instruction includes ORing the contents of said register with all binary 0 values.

14. (Currently Amended) A set of instructions residing in a storage medium, said set of instructions capable of being executed in an execution stage by a processor for implementing a method to test the processor, the method comprising:

generating a neutral instruction that causes, when executed, an architectural state value for said processor to be ascertained;

providing said neutral instruction to the execution stage of said processor; and

executing said neutral instruction to ascertain ~~an~~ said architectural state value ~~for said processor~~.

15. (Original) The set of instructions of claim 14 wherein in said method said neutral instruction is generated when a plurality of instructions are generated by a compiler.

16. (Original) The set of instructions of claim 14 wherein in said method said neutral instruction is generated by a No-operation (NOP) pseudo-random generator.

17. (Original) The set of instructions of claim 16 wherein in said method the execution of said neutral instruction causes said processor to access a value stored in a register in said processor.

18. (Original) The set of instructions of claim 14 wherein in said method the execution of said neutral instruction causes said processor to access a value stored in a register in said processor.

19. (Original) The set of instructions of claim 14 wherein in said method said neutral instruction is generated by a post-processor device.